

碩士學位論文

신원기반 암호시스템을 위한 새로운 다중  
개인키 발급모델

**A Low Cost Private Key Reissuing Model  
for Identity-based Cryptosystems**

金東玄

漢陽大學校 大學院

2005年 2月 日

碩士學位論文

신원기반 암호시스템을 위한 새로운 다중  
개인키 발급모델

**A Low Cost Private Key Reissuing Model  
for Identity-based Cryptosystems**

指導教授 吳 熙 國

이 論文을 工學碩士 學位論文으로 提出합니다.

2005年 2月 日

漢陽大學校 大學院

컴퓨터공학과

金 東 玄

이 論文을 金東玄의 碩士學位 論文으로 認准함.

2005年 2月 日

審査委員長 李廷圭 (印)

---

審査委員 許信 (印)

---

審査委員 吳熙國 (印)

---

漢陽大學校 大學院

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Pros and Cons of IBC . . . . .	1
1.2	The Inherent Key Escrow Property of IBC . . . . .	3
1.3	Multiple Private Key for Single User . . . . .	3
1.4	Private Key Revocation Problem . . . . .	4
1.5	The Need of PKI Support . . . . .	5
1.6	Our Results . . . . .	6
<b>2</b>	<b>Backgrounds</b>	<b>9</b>
<b>3</b>	<b>Related Works</b>	<b>12</b>
3.1	Boneh and Franklin’s Scheme . . . . .	12
3.2	Lee et al.’s Scheme . . . . .	14
3.3	Gentry’s Scheme . . . . .	15
3.4	Al-Riyami and Peterson’s Scheme . . . . .	17
<b>4</b>	<b>A New Private Key Issuing Model</b>	<b>20</b>
4.1	Design Rationale . . . . .	20
4.2	The Identity String . . . . .	21
4.3	The Basic Model . . . . .	22
4.4	The Advanced Key Issuing Model . . . . .	27
4.5	A New Encryption Scheme with Limited Key Escrow Capability . . . . .	30
4.6	A New Signature Scheme with Limited Key Escrow Capability . . . . .	32
<b>5</b>	<b>Analysis</b>	<b>34</b>
5.1	Comparison . . . . .	34
5.2	Security Proof of Key Issuing Protocol . . . . .	37
5.3	Security against Private Key Leakage . . . . .	43
<b>6</b>	<b>Conclusion</b>	<b>44</b>



# List of Figures

# List of Tables

1.1	Inputs used to construct a public key of a user . . . . .	5
2.1	Notations . . . . .	9
5.1	Comparison of Private Key Issuance Cost . . . . .	34



## 요약

신원기반 암호기법에서 PKG (Private Key Generator)의 권한 남용을 막기 위해 일반적으로 threshold 기법을 사용한다. 하지만 이 방법은 사용자 개인키 발급과정에서 보다 많은 인증, pairing 연산, 통신비용을 요구한다. 이 논문에서는 PKG의 권한을 분산시키는 신원기반 암호기법에서 동일한 사용자에게 다수의 개인키를 수시로 발급하거나 만료된 또는 철회된 키를 재발급하는 경우 이를 효율적으로 처리해줄 수 있는 새로운 모델을 제안한다. 새 기법에서 사용자의 개인키는 서로 다른 신뢰기관에 의해서 발급되는 두 개의 요소인 LTK (Long Term Key)와 STK (Short Term Key)로 구성된다. 이 중 LTK는 다수의 신뢰기관인 KPAs (Key Privacy Agency)들이 threshold 방법으로 발급하며, STK는 단일 신뢰기관인 KGC (Key Generation Center)가 발급한다. 이 시스템의 장점은 키 재발급 비용이 상수시간이며 공개채널을 통한 키 발급이 가능하다는 것이다. 또한 Gentry가 제안하였던 time-slot 기반의 개인키 철회기법을 다른 신원기반 암호기법보다 효율적으로 적용할 수 있다. 또한 이 논문은 제안된 키 발급 기법을 이용하여 새로운 신원기반 암호기법 및 서명기법을 소개하며, 제안된 키 발급 기법의 안전성에 대해서 증명하고 타 시스템과의 비교 분석을 통해 그 효율성을 보여준다.

# Chapter 1

## Introduction

The IBC is a public key cryptosystem where a user's public key is derived from his/her well-known ID (IDentity). In 1984, Shamir introduced the concept of IBC that includes the idea of both encryption and signature schemes [1]. However, Shamir could only realize an IBS scheme based on the RSA assumption. Until Boneh and Franklin introduced an IBE scheme based on the Weil pairing in 2001 [2], there have been no fully satisfactory solution. For example, Desmedt and Quisquater's scheme required tamper-resistant hardware [3], Tanaka's scheme required users not to collude [4], and in Maurer and Yacobi's scheme, private key generation was not practical [5]. Cocks proposed an another fully functional IBE scheme based on the quadratic residuosity assumption [6]. However, most of the current researches on IBC are based on the pairing [7, 8, 9] including our system.

### 1.1 Pros and Cons of IBC

The main concern in traditional PKI (Public Key Infrastructure) is providing a mechanism to authenticate public keys of users. To this end, a certificate, which is a digital

signature that binds a public key and its owner, is used. However, managing certificate is not just about issuing certificates. There are other things such as distribution, revocation that complicates the certificate management. In IBCs, a user's public key can be computed by anyone using the user's well-known ID such as an e-mail address without contacting its owner or a third party. As a result, public key certificates are not required in IBCs. This was the main motivation of Shamir's initial work on IBCs. Most of the problems related to certificate management in traditional PKI do not apply to IBCs. Moreover, most of the components of PKI related to certificate management can be discarded in IBCs such as public key directory. Nonetheless, IBCs cannot be considered as an alternative solution to the traditional PKI yet for the following reasons.

- As private keys of users are created by the PKG, key escrow is inherent in this system. Key escrow is an useful property that can be used to prevent crimes or recover lost keys. However, there must be a consideration about how to balance the protection of privacy of individuals with the needs of law enforcement [10].
- There is no suitable private key revocation mechanism for IBCs. Current mechanisms using CRL (Certificate Revocation List) do not suit well to IBCs. In section 1.4, we will discuss about this issue in more detail.
- A user of this system must authenticate him/herself to the PKG to obtain his/her private key, which should not be transmitted through a public channel.
- There should be an efficient way to distribute authenticated public system parameters of the PKGs. Especially, this matters more when we assume the existence of multiple PKGs that use different parameters.

## **1.2 The Inherent Key Escrow Property of IBC**

To date, several solutions to the inherent key escrow problem of IBCs have been proposed [2, 7, 9]. The most obvious solution to the key escrow problem is to use multiple PKGs in a threshold manner. Boneh and Franklin proposed this kind of system [2]. In their system, the master key, used to create private keys, is secretly shared between PKGs in a threshold manner. This approach, while resolving the key escrow problem gracefully, requires a user to contact several PKGs to obtain his/her private key. Thus, it increases the authentication, computation, and communication costs.

Gentry tried totally different approach to solve the key escrow problem [7]. He solved it by using some user chosen random secret. However, in this scheme, key escrow capability has been totally removed. Moreover, it cannot be regarded as an true identity-based scheme because a user's public key cannot be obtained directly from the ID of that user. Al-Riyami and Paterson extended Gentry's idea and provided a scheme which preserves certificateless property of IBC [9]. However, they only provided implicit authentication of the public key. Therefore, each user cannot be sure whether the public key is genuine or not.

## **1.3 Multiple Private Key for Single User**

In the initial Shamir's proposal of IBCs [1], a well-known human readable string such as e-mail address was considered as the ID of a user. Using this kind of ID as a public key has several advantages over normal public key, which is a simple binary string. First, a sender can easily obtain or already knows the ID of a receiver from which the sender can derive the authenticated public key of the receiver on his/her own. Second, one can distribute his/her ID without requiring a complicated infrastructure. Finally, as certificates contain more than just the name of the owner, ID used in IBC can contain

some additional useful information.

Boneh and Franklin proposed a form of ID which consists of an e-mail address plus some user related information such as user's duty, capability, or lifetime of a private key [2]. As a result, an ID of a user becomes more expressive, and it is possible for each user to have multiple IDs and corresponding private keys that contain identical identity information but include different additional information.

We must note that if immutable user identity information such as social security number is used solely as the ID, the master key or the user identity information must be changed when the private key has to be reissued, which are both infeasible. Moreover, in previous IBCs with limited key escrow, there is a trade off between authentication, computation, and communication costs and the degree of key escrow limit [2, 11]. Therefore, in these systems, multiple private key extraction causes much more performance degradation than the basic Boneh and Franklin's scheme [2].

## **1.4 Private Key Revocation Problem**

In traditional PKI, up-to-date CRL (Certificate Revocation List) is maintained, and users must check the status of a certificate against this list, which may be very large, before using it. This is done by making queries to a third-party or requesting the CRL itself. A revoked certificate is maintained on the CRL until the intended expiration date is over.

Gentry introduced a new time-slot approach to solve the key revocation problem in certificate-based cryptosystems [7]. This approach is based on revoking the owner of the key instead of the key itself. When a certificate is expired, the CA (Certification Authority) would issue a new certificate if and only if the owner was not revoked. However, there are two drawbacks. First, certificates of legitimate users have to be issued per each time-slot. Second, a certificate cannot be revoked in the middle of each time-slot. Gentry

Table 1.1: Inputs used to construct a public key of a user

Example	LTK	STK
1.	foo@company.com  2004	foo@company.com  2004  Role1
2.	foo@company.com  2004	foo@company.com  2004  16, Oct.
3.	foo@company.com  2004	foo@company.com  2004  Role1  16, Oct.

has ameliorated each problem using the hierarchical approach and the frequency of the certificates renewal, respectively.

In IBCs, a user private key may also need to be revoked for some reasons. Currently, there are no efficient revocation mechanisms for IBC. Trivial solution for this problem would be to use the mechanisms that are used in PKI such as CRL. However, to use mechanisms such as CRL, it requires third party queries which would offset the main advantage of IBCs of obtaining authenticated public keys of users without contacting another party.

## 1.5 The Need of PKI Support

In IBCs, the PKG must authenticate the user before issuing the user's private key. This requirement cannot be satisfied using the IBC alone. To this end, most of the IBCs assume that PKI is used for this purpose [2, 8]. Moreover, the user's private key must be transmitted securely to the user. Boneh and Franklin assume that PKI is also used for this purpose [2]. However, as shown by Lee et al. [11], simple blinding technique can be used to securely send the private key without requiring a help from other cryptosystems. Currently, the distribution of public system parameters also needs support from PKI to authenticate the parameters as well.

## 1.6 Our Results

From the previous discussion, we can conclude the followings. First, it is natural and necessary for users to have multiple private keys that are all derived from the same identity information. Second, an IBC requires mechanisms: 1) to limit the key escrow capability of the PKG, 2) to deal with revocation that does not offset its primary advantage, 3) to authenticate users before issuing private keys, 4) to securely transfer the private key to users, and 5) to distribute the authenticate system parameters.

In this paper, we introduce a novel identity-based key issuing protocol. The key issuing model limits the key escrow capability using the threshold technique as previous approaches did. However, there is no serious performance degradation when multiple private keys are issued to users. In our system, anyone can compute a user's public key without any kind of certificate, even though the private key is composed of two components which are obtained from different types of entities. The private key of a user is divided into two components: the LTK and the STK as shown in Table 1.1. The LTK is derived from the user's identity information that is normally used in IBCs. Several authorities called the KPAs, which generate LTK share after authenticating the user, issue this key in a threshold manner similar to Boneh and Franklin's method [2]. The STK is derived from the same information used to derive LTK plus some dynamic information such as a time slot or predefined role of the user. A user acquires this by contacting a single authority called the KGC. A new private key can be obtained by changing only the STK. As a result, our scheme provides a very efficient way to change or issue multiple private keys to a user.

We also give a security proof of our key issuing model, and introduce IBE and IBS schemes based on the model. We show an efficient way to adapt Gentry's revocation mechanism to IBCs as well. Although Gentry's idea can be applied to other IBCs, our

scheme requires less authentication, computation, and communication costs than others. The scheme does not assume the existence of secure channel but still assumes the existence of mechanisms supporting user authentication and distributing public system parameters.

The rest of this paper is organized as follows. In section 2, we give some definitions and explain backgrounds related to our work. In section 3, we introduce our scheme in detail. The analysis of our scheme and comparison with other schemes is given in section 4. Finally, we conclude and give some future work in section 5.



## Chapter 2

# Backgrounds

Throughout this paper, we will use the notations given in Table 2.1.

Table 2.1: Notations

Notations	Description
$q$	Large prime number
$\mathbb{G}_1$	Additive group of order $q$
$\mathbb{G}_2$	Multiplicative group of order $q$
$\mathbb{G}_1^*$	Denotes $\mathbb{G}_1/\{O\}$ , where $O$ is the identity element of $\mathbb{G}_1$
$\mathbb{Z}_q^*$	Denotes a multiplicative group of integers modulo $q$
$P$	A random generator of $\mathbb{G}_1$
$\alpha, \beta$ , and $\gamma$	Randomly distributed elements of $\mathbb{Z}_q^*$

Now, we will present some definitions related to our works.

**Definition 2.1 (Discrete Logarithm Problem (DLP) in  $\mathbb{G}_1$ ).** DLP in  $\mathbb{G}_1$  is as follow:

Given  $\langle P, \alpha P \rangle$ , acquire  $\alpha$ . An algorithm  $\mathcal{A}$  has advantage  $\varepsilon$  in solving DLP in  $\mathbb{G}_1$ , if

$$\Pr [\mathcal{A}(P, \alpha P) = \alpha] \geq \varepsilon.$$

**Definition 2.2 (Admissible Bilinear Map).**  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  is an *admissible bilinear map*, if  $\hat{e}$  has the following properties.

- **Bilinear:** Given  $P, Q, R \in \mathbb{G}_1$ , the following holds:

$$\hat{e}(P, Q+R) = \hat{e}(P, Q) \cdot \hat{e}(P, R) \text{ and } \hat{e}(P+Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R).$$

- **Non-degenerate:**  $\hat{e}(P, Q) \neq O$  for some  $P, Q \in \mathbb{G}_1$ , where  $O$  is an identity element of  $\mathbb{G}_1$ .
- **Computable:** There is an efficient algorithm to compute  $\hat{e}(P, Q)$  for any  $P, Q \in \mathbb{G}_1$ .

Bilinearity of admissible bilinear map implies  $\hat{e}(\alpha P, \beta Q) = \hat{e}(\alpha P, Q)^\beta = \hat{e}(P, \beta Q)^\alpha = \hat{e}(P, Q)^{\alpha\beta}$ . Admissible bilinear map can be constructed using the Weil or the Tate pairing on an elliptic curve over a finite field. For more detail, refer to [2].

**Definition 2.3 (Bilinear Diffie-Hellman Problem (BDHP)).** BDHP in  $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$  is as follow: Given  $\langle P, \alpha P, \beta P, \gamma P \rangle$ , compute  $\hat{e}(P, P)^{\alpha\beta\gamma}$ . An algorithm  $\mathcal{A}$  has advantage  $\varepsilon$  in solving BDHP in  $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ , if

$$\Pr [\mathcal{A}(P, \alpha P, \beta P, \gamma P) = \hat{e}(P, P)^{\alpha\beta\gamma}] \geq \varepsilon.$$

**Definition 2.4 (BDH Parameter Generator).** A randomized algorithm  $\mathcal{G}$  is called a BDH parameter generator, if it satisfies the following properties.

- It takes a single security parameter  $k \geq 1$ .
- It runs in polynomial time in  $k$ .
- It outputs the description of group  $\mathbb{G}_1$  and  $\mathbb{G}_2$  of order  $q$  and an admissible bilinear map  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ .

In this paper, we assume that the advantage on DLP and BDHP is both negligible with respect to above definitions. For more information, refer to [2].



## Chapter 3

# Related Works

In this section, we will review four schemes that are closely related to our work. In describing these schemes, we will use the following notations: 1) ID denotes the identity string of the user in concern, 2)  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$ ,  $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^l$ , and  $H_3 : \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$  denotes collision-resistant hash functions, where  $l$  is the length of the message block, and 3) ‘||’ denotes bitwise concatenation. During the system setup phase of all four schemes, an entity runs  $\mathcal{G}$  to obtain  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ . This entity also chooses a random generator  $P$  of  $\mathbb{G}_1$ .

### 3.1 Boneh and Franklin’s Scheme

Boneh and Franklin [2] introduced a  $(t, n)$  threshold based multiple authority scheme to solve the key escrow problem of a single PKG. A user of the system must contact at least  $t$  PKGs to obtain his/her private key. The protocol runs as follows.

- **System setup:** In this scheme, one of the  $n$  PKGs runs  $\mathcal{G}$ . This PKG chooses  $P \in \mathbb{G}_1^*$ ,  $H_1$ , and  $H_2$ . Each PKG generates its share of the master key  $s_i \in \mathbb{Z}_q^*$ , ( $1 \leq i \leq n$ ) in a distributed fashion using the techniques of [12]. Then each PKG

publishes its public key  $P_i = s_i P \in \mathbb{G}_1^*$ . The public parameter of this system is  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_1, \dots, P_n, P_{\text{pub}} = \sum_{i=1}^n s_i P, H_1, H_2 \rangle$ .

- **Extract:** A user sends his/her ID and authenticates him/herself to  $t$  distinct PKGs. Each PKG $_i$  computes a private key share  $d_{\text{ID}}^{(i)} = s_i Q_{\text{ID}} = s_i H_1(\text{ID}) \in \mathbb{G}_1^*$ . and returns  $d_{\text{ID}}^{(i)}$  through a secure channel to the user. The user checks the correctness of  $d_{\text{ID}}^{(i)}$  by computing the following equation:  $\hat{e}(P, d_{\text{ID}}^{(i)}) \stackrel{?}{=} \hat{e}(P_i, Q_{\text{ID}})$ .
- **Key retrieving:** When the user collects  $t$  valid shares, he/she can construct his/her private key  $d_{\text{ID}} \in \mathbb{G}_1^*$  in a threshold manner:  $d_{\text{ID}} = \sum_{i \in I} L_i s_i Q_{\text{ID}}$ , where  $L_i$  is the appropriate Lagrange coefficient. The user can also check the correctness of  $d_{\text{ID}}$  using the following equation:  $\hat{e}(P, d_{\text{ID}}) \stackrel{?}{=} \hat{e}(P_{\text{pub}}, Q_{\text{ID}})$ .
- **Encryption:** The sender encrypts the message  $m \in \{0, 1\}^l$  using the ID of the receiver by performing the following steps.
  - Computes  $Q_{\text{ID}} = H_1(\text{ID}) \in \mathbb{G}_1^*$ .
  - Chooses a random  $r \in \mathbb{Z}_q^*$ .
  - Computes  $g = \hat{e}(Q_{\text{ID}}, P_{\text{pub}})$ .
  - Computes the ciphertext  $C = \langle rP, m \oplus H_2(g^r) \rangle \in \mathbb{G}_1^* \times \{0, 1\}^l$ .
- **Decryption:** Using the private key  $d_{\text{ID}}$ , the receiver decrypts  $C = \langle U, V \rangle$  using:  $V \oplus H_2(\hat{e}(U, d_{\text{ID}})) = m$ .

A user's private key can be recovered only if more than  $t$  PKGs collude. This scheme solves the key escrow problem of single PKG gracefully. However, the authentication and communication costs between PKGs and a user have increased  $t$  times than the original scheme with single PKG. Users also have to execute  $2t + 2$  pairing computations each time to receive a private key.

### 3.2 Lee et al.'s Scheme

Lee et al. [11] also proposed a scheme using multiple authorities to solve the key escrow problem while minimizing the authentication cost. In this scheme, the KGC and  $n$  KPAs compute a user's private key in a sequential manner. The following is the detail description of Lee et al.'s protocol.

- **System setup:** In this scheme, the KGC runs  $\mathcal{G}$ . It then chooses  $P \in \mathbb{G}_1^*$ ,  $H_1$ ,  $H_2$ , and  $H_3$ . The KGC also randomly selects its master key  $s_0 \in \mathbb{Z}_q^*$  and computes its public key  $P_0 = s_0P \in \mathbb{G}_1^*$ .
- **System public key setup:** For  $1 \leq i \leq n$ , each  $\text{KPA}_i$  randomly chooses its private key  $s_i \in \mathbb{Z}_q^*$  and computes the corresponding public key  $P_i = s_iP \in \mathbb{G}_1^*$ . After that, KPAs cooperatively compute the system public key  $P_{\text{pub}} = s_0s_1 \dots s_nP \in \mathbb{G}_1^*$ . The public parameter of this system is  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_0, \dots, P_n, P_{\text{pub}}, H_1, H_2, H_3 \rangle$ .
- **Key issuing:** A user chooses a random secret  $x \in \mathbb{Z}_q^*$  and computes the blinding factor  $X = xP \in \mathbb{G}_1^*$ . Then the user sends  $X$  and ID to the KGC to request his/her partial private key. The KGC issues a blinded partial private key using the following steps.
  - **Step 1.** Verifies the user's identification.
  - **Step 2.** Sets the public key of the user as  $Q_{\text{ID}} = H_1(\text{ID} || \text{KGC} || \text{KPA}_1 || \dots || \text{KPA}_n) \in \mathbb{G}_1^*$ .
  - **Step 3.** Computes the blinded partial private key as  $Q'_0 = H_3(\hat{e}(s_0X, P_0))s_0Q_{\text{ID}} \in \mathbb{G}_1^*$  and KGC's signature on  $Q'_0$  as  $\text{Sig}_0(Q'_0) = s_0Q'_0 \in \mathbb{G}_1^*$ .
  - **Step 4.** Sends  $Q'_0$  and  $\text{Sig}_0(Q'_0)$  to the user.
- **Key securing:** For  $1 \leq i \leq n$ , the user sends ID,  $X$ ,  $Q'_{i-1}$ , and  $\text{Sig}_{i-1}(Q'_{i-1})$  to  $\text{KPA}_i$ . The recipient  $\text{KPA}_i$  performs the following steps.

- **Step 1.** Checks  $\hat{e}(\text{Sig}_{i-1}(Q'_{i-1}), P) \stackrel{?}{=} \hat{e}(Q'_{i-1}, P_{i-1})$ .
- **Step 2.** Computes  $Q'_i = H_3(\hat{e}(s_i X, P_i)) s_i Q'_{i-1} \in \mathbb{G}_1^*$  and  $\text{Sig}_i(Q'_i) = s_i Q'_i \in \mathbb{G}_1^*$ .
- **Step 3.** Sends  $Q'_i$  and  $\text{Sig}_i(Q'_i)$  to the user.

This process is repeated against all KPAs in a sequential fashion. At the end, the user obtains his/her blinded private key  $Q'_n = H_3(\hat{e}(s_n X, P_n)) s_n Q'_{n-1} \in \mathbb{G}_1^*$ .

- **Key retrieving:** The user removes the blinding factor from the blinded private key to get his/her private key  $D_{\text{ID}}$ :

$$D_{\text{ID}} = \frac{Q'_n}{H_3(\hat{e}(P_0, P_0)^x) \cdots H_3(\hat{e}(P_n, P_n)^x)} = s_0 \cdots s_n Q_{\text{ID}} \in \mathbb{G}_1^*.$$

The user can easily verify the correctness of  $D_{\text{ID}}$  by checking  $\hat{e}(D_{\text{ID}}, P) \stackrel{?}{=} \hat{e}(Q_{\text{ID}}, Y)$ .

- **Encryption/Decryption:** These procedures are identical to those of Boneh and Franklin's scheme [2].

In this scheme, a user needs to authenticate him/herself to KGC only once. Moreover, this system does not require a secure channel during the execution of the protocol. However, a user has to communicate with one KGC and every  $n$  KPAs sequentially to acquire his/her private key. The communication costs of a user have increased  $n + 1$  times than the original scheme with single PKG. Users also have to execute  $4n + 4$  pairing computations.

### 3.3 Gentry's Scheme

Gentry [7] proposed a certificate-based encryption scheme using user chosen random secret to solve the key escrow problem of a single PKG. This scheme has two types of

participant: the user and the Certificate Authority (CA). The CA plays an analogous role of the PKG in normal identity-based cryptosystem. This scheme runs as follows.

- **System setup:** In this scheme, the CA runs  $\mathcal{G}$ . Then it chooses  $P \in \mathbb{G}_1^*$ ,  $H_1$ , and  $H_2$ . It also selects its master key  $s \in \mathbb{Z}_q^*$  and sets its public key as  $P_{\text{pub}} = sP \in \mathbb{G}_1^*$ . The public parameter of this system is  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_{\text{pub}}, H_1, H_2 \rangle$ .
- **Certification:** The user requests his/her private key to the CA using the following steps.
  - **Step 1.** The user chooses a random secret  $s_U \in \mathbb{Z}_q^*$  and sets  $P_U = s_U P \in \mathbb{G}_1^*$ . He/she then sends ID and Userinfo =  $P_U || \text{ID}$  to the CA through a public channel.
  - **Step 2.** The CA checks the identification of the user and sets  $P_{\text{ID}} = H_1(P_{\text{pub}} || j || \text{Userinfo}) \in \mathbb{G}_1^*$ , where  $j$  denotes some time slot.
  - **Step 3.** The CA computes  $\text{Cert}_{\text{ID}} = sP_{\text{ID}} \in \mathbb{G}_1^*$  and returns  $\text{Cert}_{\text{ID}}$  to the user through a public channel,
  - **Step 4.** The user sets  $P'_{\text{ID}} = H_1(\text{Userinfo}) \in \mathbb{G}_1^*$  and computes his/her private key  $S_{\text{ID}} = \text{Cert}_{\text{ID}} + s_U P'_{\text{ID}}$ .
- **Encryption:** We assume that the sender and the receiver have agreed on the time slot  $j$ . A sender encrypts a message  $m$  for the user using the following step.
  - **Step 1.** Obtains Userinfo and  $\text{Cert}_{\text{ID}}$ .
  - **Step 2.** Computes  $P'_{\text{ID}} = H_1(\text{Userinfo}) \in \mathbb{G}_1^*$  and  $P_{\text{ID}} = H_1(P_{\text{pub}} || j || \text{Userinfo}) \in \mathbb{G}_1^*$ .
  - **Step 3.** Sets  $g = \hat{e}(P_{\text{pub}}, P_{\text{ID}}) \hat{e}(P_U, P'_{\text{ID}}) \in \mathbb{G}_2$  and selects a random secret  $r \in \mathbb{Z}_q^*$ .

The resulting ciphertext of  $m \in \{0, 1\}^l$  is  $C = \langle U, V \rangle = \langle rP, m \oplus H_2(g^r) \rangle \in \mathbb{G}_1^* \times \{0, 1\}^l$ .

- **Decryption:** Using the private key  $S_{ID}$ , the receiver decrypts  $C = \langle U, V \rangle$  using:  

$$V \oplus H_2(\hat{e}(U, S_{ID})) = m.$$

In this scheme, the protocol between a user and the CA is performed over a public channel. Furthermore, the communication and computation cost required during key issuing is reasonably small compared to schemes described in subsection 3.1 and 3.2. However, this scheme has removed the key escrow capability completely. Moreover, the sender cannot derive a public key directly from the public identity string of a receiver. He/she requires a certificate signed by the CA of a receiver in advance. It means this scheme has lost the main advantage of identity-based cryptosystems.

Interestingly, this scheme also deals with the revocation problem of certificate-based cryptosystems. In this protocol, a sender encrypts a message using the current time slot  $j$  without checking the current revocation status of the recipient certificate. This is possible due to the fact that if the receiver's private key has been revoked, the receiver cannot obtain his/her private key corresponding to the time slot  $j$ , preventing the decryption of the message.

### 3.4 Al-Riyami and Peterson's Scheme

Al-Riyami and Paterson [9] extended Gentry's scheme [7] to preserve certificateless property of identity-based cryptosystems. However, it only supports implicit authentication. This scheme has two types of entities: the user and the KGC, which is similar to the CA. The protocol runs as follows.

- **System setup:** This procedure is identical to that of Gentry's scheme [7]

- **Partial private key extraction:** The user sends ID to the KGC and requests his/her partial private key. The KGC checks the identity of the user and computes  $Q_U = H_1(\text{ID}) \in \mathbb{G}_1^*$ . The KGC returns  $D_U = sQ_U \in \mathbb{G}_1^*$  to the user through a secure channel. The user can verify the correctness of  $D_U$  by checking  $\hat{e}(D_U, P) \stackrel{?}{=} \hat{e}(Q_U, P_{\text{pub}})$ .
- **Secret value setup:** The user chooses a random  $x_U \in \mathbb{Z}_q^*$ .
- **Private/public key generation:** The user computes the private key  $S_U = x_U D_U = x_U s Q_U \in \mathbb{G}_1^*$  using the partial private key  $D_U$  and the user's secret value  $x_U$ . The corresponding public key is  $P_U = \langle X_U, Y_U \rangle = \langle x_U P, x_U P_{\text{pub}} = x_U s P \rangle \in \mathbb{G}_1^* \times \mathbb{G}_1^*$ .
- **Encryption:** The sender encrypts the message  $m \in \{0, 1\}^l$  using the ID and the public key  $P_U = \langle X_U, Y_U \rangle$  of the receiver by performing the following steps.
  - **Step 1.** Checks  $\hat{e}(X_U, P_{\text{pub}}) \stackrel{?}{=} \hat{e}(Y_U, P)$ .
  - **Step 2.** Computes  $Q_U = H_1(\text{ID}_A) \in \mathbb{G}_1^*$ .
  - **Step 3.** Chooses a random value  $r \in \mathbb{Z}_q^*$ .
  - **Step 4.** Computes the ciphertext  $C = \langle rP, m \oplus H_2(\hat{e}(Q_U, Y_U)^r) \rangle \in \mathbb{G}_1^* \times \{0, 1\}^l$ .
- **Decryption:** The receiver can decrypt  $C = \langle U, V \rangle$  using his/her private key  $S_U$  by computing  $V \oplus H_2(\hat{e}(S_U, U)) = m$

This scheme also exploits user chosen random secret to eliminate the key escrow problem. However, this scheme does not use any certificate. In this scheme, the sender must know the public key  $P_U$  of the receiver to encrypt a message. However, there is no way to ensure that  $P_U$  is true public key of the receiver. As a result, this protocol only supports implicit authentication. The sender is assured when the communication is successful.



## Chapter 4

# A New Private Key Issuing Model

In this section, we will describe our proposed scheme in detail. First, we will explain our design rationale. Second, we will discuss some issues concerning the structure of a user's identity string. Third, we will introduce our basic key issuing model, and then we discuss about some problems in the basic model, and present an advanced key issuing model where defects of basic model do not exist. Finally, we present new IBE and IBS schemes based on proposed key issuing model.

### 4.1 Design Rationale

The main goals of our system are the followings: 1) limit the key escrow capability, 2) provide a more efficient way to issue multiple private keys, and 3) maintain the property of IBC that public keys can be computed from well-known ID without obtaining it from the client or a third-party. With the current technology, it is difficult to limit the key escrow property without using the threshold technique. However, simply sharing the master key of the PKG in a threshold manner as done by Boneh and Franklin [2], increases the cost of private key generation. To provide a more efficient way to issue pri-

private keys, we have borrowed Gentry's idea [7]. In his scheme, the private key of a user is divided into two parts: a random value chosen by the user and a private value issued by the CA. In our scheme, the private key of a user also consists of two parts. The difference is that both parts are issued as if they are private keys of a user in an IBC to achieve the third goal of our system. However, to achieve the first goal, one of the values is issued in a threshold manner. This results in a model where there exists  $n$  authorities responsible for issuing one part of the key in a threshold manner, and a single authority responsible for issuing the other part. The benefit of this model is that a new private key can be obtained by changing one of the values making up the private key. Thus, it improves the efficiency of key issuing considerably if a user needs several different types of keys or needs to reissue his/her key frequently. In the next subsection, we discuss about the benefit of our model with respect to the structure of the identity string.

## 4.2 The Identity String

Generally, a well-known public information such as an email-address is used as the ID of a user. However, if such information is used to issue a private key of a user alone, the master key or the ID of the user must be changed when the private key has to be reissued for any reasons. We stress that both solutions are not acceptable, since the former requires all users to change their private keys and the latter may require changing well-known or immutable ID. To solve this problem, a dynamic information such as lifetime of the key is concatenated to the ID. Only the dynamic information is changed when a key has to be reissued.

As mentioned above, applying simple threshold technique to solve the inherent key escrow problem of IBC causes additional costs during the key issuing phase. The more private key is issued, the worse this drawback is. In our system, we provide a more

efficient way of reissuing private keys by dividing the key into two components: LTK and STK . The LTK is derived from the ID of a user and the lifetime of LTK. This key is obtained in a threshold manner. The STK is derived from the same information used to derive the corresponding LTK plus some dynamic information. Dynamic information can be time-slot information like Gentry’s scheme [7], a role of the user, or the usage of the key. Therefore, it is reasonable to assume that there are some rules to determine the dynamic information of intended user. For example, one can determine the current time-slot using some predefined rules and the current time, or one can already know the role of the receiver. The STK is obtained from a single authority. The same information used to derive the LTK must be used as one of the input used to derive the corresponding STKs to bind the two keys. This binding is required to limit the number of private keys of a user to those that are actually issued. Examples of inputs used to construct the corresponding public key of a user are given in Table 1.1.

A new private key can be issued by changing either the LTK or the STK. More precisely, the private key can be reissued only by changing the STK. If the LTK needs to be reissued, every related STKs must be discarded. Since, the STK can be obtained from a single authority, our system provides a very efficient way of reissuing private keys.

### 4.3 The Basic Model

In this section, we introduce our basic key issuing model. Before proceeding in detail, we will introduce the participants of our basic model and briefly explain their roles. There are three types of entities in the model:  $n$  KPAs, the KGC, and users. The role of each entity is as follows.

- **KPAs:** KPAs are responsible for issuing LTK shares for users. A user has to contact at least  $t$  KPAs of total  $n$  KPAs to obtain his/her LTK. A user receives

a share of his/her LTK from each KPA through a public channel and constructs his/her LTK in a threshold manner.

- **KGC:** KGC is responsible for issuing STKs for users. A user sends his/her identity, the blinded LTK, and dynamic short-term information to obtain the STK. The STK is issued through a public channel.
- **User:** A user needs to obtain his/her LTK shares from  $t$  KPAs and his/her STK from the KGC to construct his/her private key of the system.

Now, we introduce our basic key issuing model, which consists of the following two procedures: system setup and user private key extraction.

### System Setup

In this phase, each authority determines its public/private keys and computes public system parameters cooperatively. This procedure consists of two sub procedures: KGC setup and KPAs setup.

- **KGC setup:** The KGC carries out the following procedures.
  - **Step 1.** Runs the bilinear parameter generator  $\mathcal{G}$  and obtains  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ .
  - **Step 2.** Chooses a random generator  $P$  of  $\mathbb{G}_1$ .
  - **Step 3.** Selects cryptographic hash functions

$$H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^* \text{ and } H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^l,$$

where  $l$  is the length of the message block.

- **Step 4.** Selects its master key  $s \in \mathbb{Z}_q^*$  and sets  $P_{\text{KGC}} = sP \in \mathbb{G}_1^*$ .
- **KPAs setup:** Each KPAs computes its own private and public keys. They also compute the private key and public key of KPAs cooperatively.

- **Step 1.** Each  $KPA_i$  generates its master key  $x_i \in \mathbb{Z}_q^*$ , ( $1 \leq i \leq n$ ) in a distributed fashion using the techniques of [12]. It also computes its public key  $P_{KPA_i} = x_i P \in \mathbb{G}_1^*$ .
- **Step 2.** The private key of KPAs is  $x = \sum_{i \in I} L_i x_i \in \mathbb{Z}_q^*$ , where  $L_i$  is appropriate Lagrange coefficient and  $I$  is a set of  $t$  numbers which are greater than or equal to 0 and less than or equal to  $n$ . The corresponding public key of KPAs is  $P_{KPAs} = xP \in \mathbb{G}_1^*$ .
- The public parameters of this system is  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_{KGC}, P_{KPAs}, P_{KPA_1}, \dots, P_{KPA_n}, H_1, H_2 \rangle$ .

### User Private Key Extraction

A user acquires his/her private key using this procedure. This procedure is divided into the following three sub procedures: LTK extraction, STK extraction, and user private key computation.

- **LTK extraction:** The user requests his/her LTK shares to  $t$  different  $KPA_i$ , ( $1 \leq i \leq n$ ) using the following steps. We must note that the user has to authenticate him/herself to each KPA involved in this procedure before proceeding the step 1.
  - **Step 1.** The user chooses his/her random blinding factor  $b_i \in \mathbb{Z}_q^*$  and computes  $b_i Q_{ID} = b_i H_1(ID)$ .
  - **Step 2.** The user sends  $\langle ID, b_i P, b_i Q_{ID} \rangle$  to  $KPA_i$ .
  - **Step 3.** The  $KPA_i$  checks the identification of the user and verifies the validness of  $b_i Q_{ID}$  using the following equation:

$$\hat{e}(Q_{ID}, b_i P) = \hat{e}(Q_{ID}, P)^{b_i} \stackrel{?}{=} \hat{e}(b_i Q_{ID}, P) = \hat{e}(Q_{ID}, P)^{b_i}. \quad (4.1)$$

- **Step 4.** The  $KPA_i$  computes the blinded LTK share  $b_i d_{ID}^{(i)} = b_i x_i Q_{ID} \in \mathbb{G}_1^*$ . Finally, it returns  $b_i d_{ID}^{(i)}$  to the user.
- **Step 5.** The user removes the blind factor by computing  $b_i^{-1} b_i d_{ID}^{(i)} = d_{ID}^{(i)}$  and verifies the correctness of  $d_{ID}^{(i)} = x_i Q_{ID}$  using the following equation:

$$\hat{e}(P, d_{ID}^{(i)}) = \hat{e}(P, x_i Q_{ID}) = \hat{e}(P, Q_{ID})^{x_i} \stackrel{?}{=} \hat{e}(P_{KPA_i}, Q_{ID}) = \hat{e}(x_i P, Q_{ID}) = \hat{e}(P, Q_{ID})^{x_i}. \quad (4.2)$$

After collecting  $t$  different LTK shares, the user computes his/her LTK:

$$d_{ID} = \sum_{i \in I} L_i d_{ID}^{(i)} = \sum_{i \in I} L_i x_i Q_{ID} \in \mathbb{G}_1^*,$$

where  $L_i$  is the appropriate Lagrange coefficient. The user can check the correctness of his/her LTK using the following equation:

$$\hat{e}(P, d_{ID}) = \hat{e}(P, x Q_{ID}) = \hat{e}(P, Q_{ID})^x \stackrel{?}{=} \hat{e}(P_{KPAS}, Q_{ID}) = \hat{e}(x P, Q_{ID}) = \hat{e}(P, Q_{ID})^x.$$

- **STK extraction:** The user requests the STK by sending  $\langle ID, T, X = b P_{KPAS}, Y = b d_{ID}, Z = b Q_{(ID, T)} \rangle$  to the KGC, where  $b \in \mathbb{Z}_q^*$  is a random blinding factor chosen by the user and  $T$  is the dynamic user information that will be used to derive the STK. The KGC would reject the request of the user, if the user's LTK  $d_{ID}$  has been revoked. Otherwise, the KGC computes requested STK, and sends it back to the user.

- **Step 1.** The KGC checks whether the user has a valid LTK using the follow-

ing equation:

$$\begin{aligned}\hat{e}(Q_{\text{ID}}, X) &= \hat{e}(Q_{\text{ID}}, bP_{\text{KPAs}}) = \hat{e}(Q_{\text{ID}}, bxP) = \hat{e}(Q_{\text{ID}}, P)^{bx} \\ &\stackrel{?}{=} \hat{e}(P, Y) = \hat{e}(P, bd_{\text{ID}}) = \hat{e}(P, bxQ_{\text{ID}}) = \hat{e}(P, Q_{\text{ID}})^{bx} = \hat{e}(Q_{\text{ID}}, P)^{bx}.\end{aligned}$$

- **Step 2.** The KGC sets  $Q_{\langle \text{ID}, T \rangle} = H_1(\text{ID}||T)$  and checks whether the user has sent a valid  $Z$  using the following equation:

$$\begin{aligned}\hat{e}(Q_{\langle \text{ID}, T \rangle}, X) &= \hat{e}(Q_{\langle \text{ID}, T \rangle}, bP_{\text{KPAs}}) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}})^b \\ &\stackrel{?}{=} \hat{e}(Z, P_{\text{KPAs}}) = \hat{e}(bQ_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}}) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}})^b.\end{aligned}$$

- **Step 3.** The KGC sends  $sZ$  to the user.
- **Step 4.** The user computes his/her STK  $d_{\langle \text{ID}, T \rangle}$  by eliminating blinding factor as follows:  $b^{-1}sZ = b^{-1}sbQ_{\langle \text{ID}, T \rangle} = sQ_{\langle \text{ID}, T \rangle} = d_{\langle \text{ID}, T \rangle}$ .
- **Step 5.** The user checks the correctness of  $d_{\langle \text{ID}, T \rangle}$  using the following equation:

$$\begin{aligned}\hat{e}(P, d_{\langle \text{ID}, T \rangle}) &= \hat{e}(P, sQ_{\langle \text{ID}, T \rangle}) = \hat{e}(P, Q_{\langle \text{ID}, T \rangle})^s \\ &\stackrel{?}{=} \hat{e}(P_{\text{KGC}}, Q_{\langle \text{ID}, T \rangle}) = \hat{e}(sP, Q_{\langle \text{ID}, T \rangle}) = \hat{e}(P, Q_{\langle \text{ID}, T \rangle})^s.\end{aligned}$$

- **User private key computation:** The user computes his/her private key  $D_{\langle \text{ID}, T \rangle} = d_{\text{ID}} + d_{\langle \text{ID}, T \rangle} \in \mathbb{G}_1$ . If the computed  $D_{\langle \text{ID}, T \rangle}$  is the identity element of  $\mathbb{G}_1$ , we need to change the ID appropriately. However, this possibility is negligible. Therefore, we assume  $D_{\langle \text{ID}, T \rangle} \in \mathbb{G}_1^*$ .

## 4.4 The Advanced Key Issuing Model

We have introduced a new key issuing model which not only limits key escrow property but also support efficient multiple private key extractions. However, more than  $t$  KPAs cannot escrow a private key of a user in our basic model. More specifically, even though more than  $t$  KPAs can recover a LTK in a threshold manner, escrowing a STK requires the cooperation of KGC. This means that the basic model does not provide the threshold property.

In this section, we present a more advanced key issuing model that provides the threshold property as well as maintains every positive aspect of the basic model. To support the threshold property, we verifiably encrypt all the issued STK using the public key of KPAs and publish these encryptions. Therefore, more than  $t$  KPAs can cooperate to obtain the STK required to escrow a private key of a user. However, since we cannot trust that the KGC will always commit the verifiable encrypted STK correctly, we introduce another authority called the KCA (Key Commitment Agency) that observes KGC's activity. To enforce this observation, we use the KCA as an intermediary between the user and the KGC. We also assume that the KCA monitors KGC's activity, which prevents users directly contacting the KGC. As a result, unless KGC and KCA collude, there would be a publicly available verifiable encrypted STK corresponding to every STK issued by the KGC. In this model, users cannot directly contact the KGC. They must request the STK to the KCA, and the KCA will acquire the blinded STK from the KGC in behalf of the user. In summary, the roles of KCA are as follows: 1) forwards user's STK issuance request to the KGC, 2) receives KGC's response, which includes blinded STK and verifiably encrypted STK, and 3) sends blinded STK back to the user and publishes verifiably encrypted STK. Now, we will explain the procedures of the advanced model.

## System Setup

This phase is the same as that of the basic model's. The resulting public parameters is

$$\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_{KGC}, P_{KPAs}, P_{KPA_1}, \dots, P_{KPA_n}, H_1, H_2 \rangle.$$

## User Private Key Extraction

A user acquires his/her private key through this procedure, which consist of the following two sub procedures: LTK extraction and STK extraction.

- **LTK extraction:** This phase is also identical to that of the basic model's. After this phase, a user can compute his/her LTK  $d_{ID}$ .
- **STK extraction:** The user requests the STK by sending  $\langle ID, T, X = bP_{KPAs}, Y = bd_{ID}, Z = bQ_{\langle ID, T \rangle} \rangle$  to the KCA, where  $b \in \mathbb{Z}_q^*$  is a random blinding factor chosen by the user. The user would get his/her STK after successfully proceeding the following three procedures.
  - **Request forwarding:** The KCA forwards the request message of the user to the KGC without modifying it.
  - **STK issuance:** The KGC would reject the request of the user, if the user's LTK  $d_{ID}$  has been revoked. Otherwise, the KGC computes requested STK, and sends it back to the KCA.

- **Step 1.** The KGC checks whether the user has a valid LTK using the following equation:

$$\begin{aligned} \hat{e}(Q_{ID}, X) &= \hat{e}(Q_{ID}, bP_{KPAs}) = \hat{e}(Q_{ID}, bxP) = \hat{e}(Q_{ID}, P)^{bx} \\ &\stackrel{?}{=} \hat{e}(P, Y) = \hat{e}(P, bd_{ID}) = \hat{e}(P, bxQ_{ID}) = \hat{e}(P, Q_{ID})^{bx} = \hat{e}(Q_{ID}, P)^{bx}. \end{aligned} \tag{4.3}$$

- **Step 2.** The KGC sets  $Q_{\langle \text{ID}, T \rangle} = H_1(\text{ID}||T)$  and checks whether the user has sent a valid  $Z$  using the following equation:

$$\begin{aligned} \hat{e}(Q_{\langle \text{ID}, T \rangle}, X) &= \hat{e}(Q_{\langle \text{ID}, T \rangle}, bP_{\text{KPAs}}) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}})^b \\ &\stackrel{?}{=} \hat{e}(Z, P_{\text{KPAs}}) = \hat{e}(bQ_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}}) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P_{\text{KPAs}})^b. \end{aligned} \quad (4.4)$$

- **Step 3.** The KGC sends the blinded STK  $sZ$  to the KCA. It also transmits the verifiably encrypted STK  $\langle V = sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}}, \bar{V} = \bar{b}P \rangle$ , where  $\bar{b}$  is a random element of  $\mathbb{Z}_q^*$  to the KCA.

- **STK commitment:** The KCA checks the correctness of  $\langle V, \bar{V} \rangle$  using the prior registered request as follows:

$$\begin{aligned} \hat{e}(V, P) &= \hat{e}(sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}}, P) = \hat{e}(sQ_{\langle \text{ID}, T \rangle} + \bar{b}xP, P) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P)^s \hat{e}(P, P)^{\bar{b}x} \\ &\stackrel{?}{=} \hat{e}(Q_{\langle \text{ID}, T \rangle}, P_{\text{KGC}}) \hat{e}(\bar{V}, P_{\text{KPAs}}) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, sP) \hat{e}(\bar{b}P, xP) = \hat{e}(Q_{\langle \text{ID}, T \rangle}, P)^s \hat{e}(P, P)^{\bar{b}x}. \end{aligned} \quad (4.5)$$

If  $\langle V, \bar{V} \rangle$  pair is correct, the KCA publishes  $\langle \text{ID}, T, V, \bar{V} \rangle$  and sends  $sZ$  to the user. Interestingly, any third party can also to check the correctness of the verifiably encrypted STK  $\langle V, \bar{V} \rangle$ .

- **STK computation:** The user computes his/her STK  $d_{\langle \text{ID}, T \rangle}$  by eliminating the blinding factor as follows:  $b^{-1}sZ = b^{-1}sbQ_{\langle \text{ID}, T \rangle} = sQ_{\langle \text{ID}, T \rangle} = d_{\langle \text{ID}, T \rangle}$ . The user also checks the correctness of  $d_{\langle \text{ID}, T \rangle}$  using the following:

$$\begin{aligned} \hat{e}(P, d_{\langle \text{ID}, T \rangle}) &= \hat{e}(P, sQ_{\langle \text{ID}, T \rangle}) = \hat{e}(P, Q_{\langle \text{ID}, T \rangle})^s \\ &\stackrel{?}{=} \hat{e}(P_{\text{KGC}}, Q_{\langle \text{ID}, T \rangle}) = \hat{e}(sP, Q_{\langle \text{ID}, T \rangle}) = \hat{e}(P, Q_{\langle \text{ID}, T \rangle})^s. \end{aligned} \quad (4.6)$$

- **User private key computation:** The user computes his/her private key  $D_{\langle \text{ID}, T \rangle} = d_{\text{ID}} + d_{\langle \text{ID}, T \rangle} \in \mathbb{G}_1$ .

### Key Escrow

In the proposed scheme, the TTP (Trusted Third Party) which already knows  $\langle \text{ID}, T, V, \bar{V} \rangle$ , can recover a private key of a user by cooperating with more than  $t$  KPAs as follows.

- The TTP sends  $\bar{V} = \bar{b}P$  to more than  $t$  KPAs. Each  $\text{KPA}_i$  computes  $\bar{V}^{(i)} = x_i \bar{V} = x_i \bar{b}P$ , and sends it back to the TTP. The TTP then checks the validity of each  $\bar{V}^{(i)}$  as follows:

$$\hat{e}(\bar{V}, P_{\text{KPA}_i}) = \hat{e}(\bar{b}P, x_i P) = \hat{e}(P, P)^{\bar{b}x_i} \stackrel{?}{=} \hat{e}(\bar{V}^{(i)}, P) = \hat{e}(x_i \bar{b}P, P) = \hat{e}(P, P)^{\bar{b}x_i}.$$

- The TTP computes  $\tilde{V} = \sum_{i \in I} L_i \bar{V}^{(i)} = \sum_{i \in I} L_i x_i \bar{b}P = \bar{b}xP = \bar{b}P_{\text{KPAs}}$ , and recovers the STK of a user as following:

$$V - \tilde{V} = sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}} - \bar{b}P_{\text{KPAs}} = sQ_{\langle \text{ID}, T \rangle} = d_{\langle \text{ID}, T \rangle}.$$

- The TTP also requests LTK share of the user to more than  $t$  KPAs. This procedure is an analogous of LTK extraction phase. Finally, the TTP recovers the private key of the user  $D_{\langle \text{ID}, T \rangle} = d_{\text{ID}} + d_{\langle \text{ID}, T \rangle}$ .

## 4.5 A New Encryption Scheme with Limited Key Escrow Capability

In this section, we present a new IBE scheme evolved from Boneh and Franklin's IBE [2]. Private keys used in this encryption scheme are issued by using either our basic model or advanced one. In the proposed IBE scheme, a public key of a receiver is derived from  $\langle \text{ID}, T \rangle$  pair. The protocol runs as follows.

## Encryption

Let  $T$  be the dynamic information used in computing the STK of the user. We assume that the sender and the receiver have agreed on  $T$  in advance. The sender obtains the public key of the user by computing  $Q_{\text{ID}} = H_1(\text{ID}) \in \mathbb{G}_1^*$  and  $Q_{\langle \text{ID}, T \rangle} = H_1(\text{ID}||T) \in \mathbb{G}_1^*$ . Then the sender sets

$$g = \hat{e}(P_{\text{KPAS}}, Q_{\text{ID}}) \hat{e}(P_{\text{KGC}}, Q_{\langle \text{ID}, T \rangle}) \in \mathbb{G}_2$$

and selects a random secret  $r \in \mathbb{Z}_q^*$ . The resulting ciphertext of a message  $m \in \{0, 1\}^l$  is

$$C = \langle rP, m \oplus H_2(g^r) \rangle \in \mathbb{G}_1^* \times \{0, 1\}^l.$$

## Decryption

The receiver can decrypt  $C = \langle U, V \rangle$  using his/her private key  $D_{\langle \text{ID}, T \rangle}$ :

$$m = V \oplus (H_2(\hat{e}(U, D_{\langle \text{ID}, T \rangle}))).$$

The correctness of the encryption scheme can be easily verified as follows:

$$\begin{aligned} \hat{e}(U, D_{\langle \text{ID}, T \rangle}) &= \hat{e}(U, xQ_{\text{ID}} + sQ_{\langle \text{ID}, T \rangle}) &= \hat{e}(U, xQ_{\text{ID}}) \hat{e}(U, sQ_{\langle \text{ID}, T \rangle}) \\ &= \hat{e}(rP, xQ_{\text{ID}}) \hat{e}(rP, sQ_{\langle \text{ID}, T \rangle}) &= \hat{e}(P, Q_{\text{ID}})^{xr} \hat{e}(P, Q_{\langle \text{ID}, T \rangle})^{sr} \\ &= \hat{e}(xP, Q_{\text{ID}})^r \hat{e}(sP, Q_{\langle \text{ID}, T \rangle})^r &= \hat{e}(P_{\text{KPAS}}, Q_{\text{ID}})^r \hat{e}(P_{\text{KGC}}, Q_{\langle \text{ID}, T \rangle})^r = g^r. \end{aligned}$$

## 4.6 A New Signature Scheme with Limited Key Escrow Capability

In 2003, Hess proposed an IBS scheme from pairing [8]. In this scheme, a signer can make a signature of a message using his/her private key, and anyone can check the correctness of the signature using an ID of the signer. Now, we introduce another version of Hess's IBS scheme, where a private key of a user is issued through either our basic model or advanced one. The protocol runs as follows.

### Signature Generation

A signer having his/her private key  $D_{\langle \text{ID}, T \rangle}$  makes a signature of a message  $m$  as follows.

- Computes  $k = \hat{e}(P_r, P)^r$ , where  $P_r$  and  $r$  are random elements of  $\mathbb{G}_1^*$  and  $\mathbb{Z}_q^*$  respectively.
- Resulting signature of a message  $m$  is  $\langle v = H_2(m, k), u = vD_{\langle \text{ID}, T \rangle} + rP_r \rangle$ .

### Verification

To verify a signature  $\langle u, v \rangle$  of a message  $m$ , the verifier first computes  $k$  as follows:

$$\begin{aligned}
 k &= \hat{e}(u, P) \hat{e}(Q_{\text{ID}}, -P_{\text{KPAs}})^v \hat{e}(Q_{\langle \text{ID}, T \rangle}, -P_{\text{KGC}})^v \\
 &= \hat{e}(vD_{\langle \text{ID}, T \rangle} + rP_r, P) \hat{e}(Q_{\text{ID}}, -P_{\text{KPAs}})^v \hat{e}(Q_{\langle \text{ID}, T \rangle}, -P_{\text{KGC}})^v \\
 &= \hat{e}(vD_{\langle \text{ID}, T \rangle} + vD_{\text{ID}} + rP_r, P) \hat{e}(Q_{\text{ID}}, -P_{\text{KPAs}})^v \hat{e}(Q_{\langle \text{ID}, T \rangle}, -P_{\text{KGC}})^v \\
 &= \hat{e}(vS Q_{\langle \text{ID}, T \rangle} + vX Q_{\text{ID}} + rP_r, P) \hat{e}(Q_{\text{ID}}, -XP)^v \hat{e}(Q_{\langle \text{ID}, T \rangle}, -SP)^v \\
 &= \hat{e}(Q_{\langle \text{ID}, T \rangle}, P)^{vS} \hat{e}(Q_{\text{ID}}, P)^{vX} \hat{e}(P_r, P)^r \hat{e}(Q_{\text{ID}}, P)^{-vX} \hat{e}(Q_{\langle \text{ID}, T \rangle}, P)^{-vS} \\
 &= \hat{e}(P_r, P)^r,
 \end{aligned}$$

and checks the following equation:

$$v \stackrel{?}{=} H_2(m, k).$$

# Chapter 5

## Analysis

### 5.1 Comparison

In this section, we compare our advanced key issuing model against the schemes proposed by Boneh and Franklin [2] and Lee et al. [11], which are IBCs that have a limited key escrow property. Especially, we compare the following aspects: 1) efficiency of key issuance, and 2) key escrow procedure.

Table 5.1: Comparison of Private Key Issuance Cost

	Initial Issuance Cost			Reissuance Cost		
	[2]	[11]	Ours	[2]	[11]	Ours
Pairing computation	$2t + 2^*$	$4t + 4$	$(4t + 4) + 9$	$2t + 2^*$	$4t + 4$	9
Communication	$2t$	$2t + 2$	$2t + 4$	$2t$	$2t + 2$	4
Authentication	$t$	1	$t$	$t$	1	0

$t$  denotes the number of PKGs participating in [2] and the number of KPAs participating in [11] and ours.

\* [2] requires secure channels whereas [11] and ours do not. If the method used in our system to remove the secure channel assumption to [2], the cost becomes  $4t + 2$ .

## Efficiency of Key Issuance

Before proceeding in detail, we note that [2] and ours are based on threshold technique but [11] isn't. Moreover, [2] and ours can communicate in parallel while [11] must do in a sequential manner. This means that as the number of authorities increases, [11] will require more communication time. We must note that we must consider the extra communication time due to using the KCA as an intermediary between the user and the KGC in our scheme.

For more precise and fair comparison, we assume that the threshold variable  $t$  of [2] and ours is equal to the number of KPAs in [11]. In other words, we consider the setting where the number of authorities used in ours is not only equal to that of [11], but one more than that of [2] without considering the KCA. Now, we will compare each key issuing model based on each of the following aspects: pairing computation, communication, and authentication.

- **Pairing computation:** As shown in Table 5.1, [2] requires the least pairing computations during the initial key issuance. However if the method used in our scheme to remove the secure channel assumption is applied to [2], the computation cost of [2] becomes  $4t + 2$ . In this setting, we can conclude that ours requires 11 more pairing computation than [2] and 9 more pairing computation than [11] during the initial key issuance. These additional computations are due to the STK extraction and checking the validity of the verifiable encrypted STK by the KCA. However, in our scheme, the computation cost of reissuing a private key is constant.
- **Communication:** [2] requires the least communication cost. Our protocol requires 4 more communication than [2] which is due to the STK extraction where the KCA plays an intermediary role between the user and the KGC. Additionally,

[11] will need extra communication time because of its sequential property. In our scheme, the communication cost of reissuing a private key is also constant.

- **Authentication:** [11] needs the least authentication. However, in our scheme, authentication is not required when key is being reissued. From these, we can conclude that our scheme outperforms others with respect to reissuance, and performs nearly equal to others with respect to initial issuance.

We must note that in our scheme if the LTK of a user must be reissued for any reason the cost of reissuing the private key is identical to that of initial key issuance.

### **Key Escrow Procedure**

Finally, we discuss about the key escrow procedure. As mentioned before, all three systems have key escrow property. In [2], key escrow procedure is done by doing usual secret recovering protocol in the threshold scheme which does not require pairing computation. Our scheme requires twice as much amount of computation as [2] does assuming that the committed STK is known. This is because in our system, each  $t$  KPA compute LTK share and STK recovery share to escrow the private key of a user. Since ours and [2] are based on the traditional threshold concept, any  $t$  out of  $n$  authorities can recover the key. This procedure does not require sequential connection. On the other hand, in [11], the private key is recovered in a sequential manner by all the authorities originally participated in the key issuing protocol. This means that [11] needs relatively more time to recover the private key than others and all authorities must participate in this procedure.

## 5.2 Security Proof of Key Issuing Protocol

In this section, we enumerate every possible types of adversaries in our advanced key issuing model and analyze their ability. Then, we define security requirements of our key issuing model with some assumptions. Finally, we show that the model satisfies security requirements based on the assumptions.

### The Adversaries

There are four types of entities in our advanced key issuing model:  $n$  KPAs, KGC, KCA, and users, all of whom we can consider as latent adversaries. We also have to consider a malicious third party, which does not participate in the key issuing procedure as a legitimate party, as a potential adversary. Now, we scrutinize each adversary model and define their ability.

- **Malicious third party:** An adversary that does not participate as one of the legal participants are called malicious third party. These adversaries can eavesdrop or modify messages which are sent through public communication channel between legitimate parties. The goal of these adversary is as follows: 1) disturb the key issuing phase, 2) acquire a LTK of a user corresponding to an ID, 3) obtain a STK of a user related to  $\langle ID, T \rangle$ , and acquire the private key of a user.
- **KCA:** The KCA is supplementary authority supporting threshold property in our key issuing model. It not only forwards user request to the KGC, but also sends the response from KGC back to the user, as well as publishes verifiably encrypted STK. However, as each message passing through the KCA is blinded, KCA does not have any explicit knowledge about STK or LTK. As a result, the KCA only has ability as much as the malicious third party does.

- **KPAs:** As mentioned before, more than or equal to  $t$  KPAs can compute correct LTK corresponding to the ID in a threshold manner. However, if the threshold cryptosystem is correct and secure, it is infeasible for less than  $t$  KPAs to compute the correct LTK corresponding to the ID. In this setting, each KPA cannot compute the LTK using its own master secret share. Therefore, in this respect, although they can compute a share of the LTK, they have the same amount of ability as the malicious third party does.
- **KGC:** In our model, the KGC knows the master key  $s$  and can compute STKs corresponding to any  $\langle \text{ID}, T \rangle$  pairs of users. This means that the KGC not only has ability as much as the malicious third party does but also can compute any STK whichever it wants.
- **Users:** Throughout proposed protocol, a legitimate user can possess correct STKs and LTKs related to his/her own  $\langle \text{ID}, T \rangle$  pairs. This is because that a user is allowed to request his/her LTKs or STKs to the KPAs or KGC respectively, even though a user does not participate in key issuing process directly. In addition to this, a user can do whatever the malicious third party can.

## Security Requirements

From previous discussion, we can conclude the followings: 1) proposed key issuing protocol must not suffer from message modification, insertion or replay attack, 2) the LTK can be computed only if more than  $t$  KPAs cooperate, 3) the KGC has to be the unique entity which can compute the correct STK, and 4) except legitimate user, no other entities know any private key of the user. From these conclusions, we now define the security requirements of our key issuing protocol.

**Definition 5.1.** The security requirements of advanced key issuing protocol is as follows.

- **Correctness:** Incorrect response from any entity, message modification during transmission, replying attack by using old messages, or incorrect packet insertion cannot make any harm to our protocol. Especially a user must be able to determine whether a private key, issued through our key issuing model, is correct or not.
- **Unforgeability:** It has to be impossible to compute correct LTK related to an ID without cooperation of more than  $t$  KPAs. Moreover, the KGC is the only entity that can compute the correct STK of  $\langle \text{ID}, T \rangle$ .
- **Privacy:** Except a user who possesses  $\langle \text{ID}, T \rangle$ , no other entities can have a correct private key corresponding to  $\langle \text{ID}, T \rangle$  unless more than  $t$  KPAs cooperate.

## Security Proof

Before proceeding in detail, we introduce some assumptions which are used to prove the security of our system, and then give the security proofs of our model.

**Assumption 5.1.** The threshold scheme used in our system is secure in that it is computationally infeasible to obtain the shared secret when less than  $t$  KPAs collude.

For more detail information on the security of the threshold technique, refer to [12].

**Assumption 5.2.** Advantage of an adversary on DLP in  $\mathbb{G}_1$  is negligible.

**Assumption 5.3.** Following public information is well-known to all entities in our key issuing model.

- The public system parameters:  $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, l, P, P_{\text{KGC}}, P_{\text{KPAs}}, P_{\text{KPA}_1}, \dots, P_{\text{KPA}_n}, H_1, H_2 \rangle$ .
- $\langle \text{ID}, b_i P, b_i Q_{\text{ID}} \rangle$  and  $b_i d_{\text{ID}}^{(i)} = b_i x_i Q_{\text{ID}}$  pairs, which are published during LTK issuance phase.

- $\langle \text{ID}, T, X = bP_{\text{KPAs}}, Y = bd_{\text{ID}}, Z = bQ_{\langle \text{ID}, T \rangle} \rangle$ , and  $sZ = bd_{\langle \text{ID}, T \rangle}$ ,  $\langle \text{ID}, T, V = sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}}, \bar{V} = \bar{b}P \rangle$  pairs, which are known publicly during STK issuance phase.

As we have discussed before, there are five types of potential adversaries in our protocol. However, an ability of the malicious third party, the KCA, and less than  $t$  KPAs are equal based on Assumption 5.1. Furthermore, we can conclude that both the KGC, which can compute any STK whichever it wishes, and users, which already know lots of LTKs and STKs of their own, are more powerful adversaries than malicious third party, the KCA, or less than  $t$  KPAs. Therefore, our protocol would be secure enough if it were secure against the KGC, users or even their collusion. Now, we present two lemmas and their proofs. Then, we show that the key issuing model satisfies our security requirements based on those lemmas.

**Lemma 5.1.** Proposed key issuing model satisfies the correctness requirement.

*proof.* In LTK extraction phase, a user sends  $\langle \text{ID}, b_iP, b_iQ_{\text{ID}} \rangle$  to  $t$  KPAs. Then each  $\text{KPA}_i$  1) authenticates the user, 2) computes LTK share  $b_i d_{\text{ID}}$ , and 3) sends  $b_i d_{\text{ID}}$  back to the user. The correctness of  $\langle \text{ID}, b_iP, b_iQ_{\text{ID}} \rangle$  can be verified using the Equation 4.1. The correctness of  $b_i d_{\text{ID}}$  can be verified using the Equation 4.2 as well.

In STK extraction phase, a user sends  $\langle \text{ID}, T, X = bP_{\text{KPAs}}, Y = bd_{\text{ID}}, Z = bQ_{\langle \text{ID}, T \rangle} \rangle$  to the KGC through the KCA. Then, the KGC sends  $bd_{\langle \text{ID}, T \rangle}$  and  $\langle \text{ID}, T, V = sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}}, \bar{V} = \bar{b}P \rangle$  back to the KCA. Finally, the KCA sends  $bd_{\langle \text{ID}, T \rangle}$  back to the user. Using the Equation 4.3 and 4.4, the KGC can check the correctness of  $\langle \text{ID}, T, X = bP_{\text{KPAs}}, Y = bd_{\text{ID}}, Z = bQ_{\langle \text{ID}, T \rangle} \rangle$ . The KCA is also able to verify the correctness of  $\langle \text{ID}, T, V = sQ_{\langle \text{ID}, T \rangle} + \bar{b}P_{\text{KPAs}}, \bar{V} = \bar{b}P \rangle$  using the Equation 4.5. Finally, the user can verify the correctness of his/her STK  $d_{\langle \text{ID}, T \rangle}$  using the Equation 4.6.

An adversary may disturb the LTK issuance by intercepting  $\langle \text{ID}, b_iP, b_iQ_{\text{ID}} \rangle$  and sending  $\langle \text{ID}, n_a b_iP, n_a b_iQ_{\text{ID}} \rangle$  instead using a random value  $n_a \in \mathbb{Z}_q^*$ . Although each KPA

cannot detect this modification using the Equation 4.1, the user can still check the correctness of the resulting LTK share by using the Equation 4.2. Therefore, the user cannot be disturbed by this attack during the LTK issuance. Similarly an adversary may hinder the STK issuance by intercepting  $\langle \text{ID}, T, X, Y, Z \rangle$  and sending  $\langle \text{ID}, T, n_a X, n_a Y, n_a Z \rangle$  instead using a random value  $n_a \in \mathbb{Z}_q^*$ . Even though the KGC cannot detect this change using the Equation 4.3 and 4.4, the user can still verify the validity of resulting STK by using the Equation 4.6. Therefore, the user cannot be disturbed by this attack during STK issuance.

By summing up the discussions mentioned above, we can conclude that each entity can check the correctness of messages which they have received from others, even though there are two exceptions which make no harm to our protocol. A user is also able to check the validity of LTK and STK which he/she received through the key issuing model. Therefore, the lemma holds.  $\square$

**Lemma 5.2.** Proposed key issuing model satisfies both the unforgeability and privacy requirements.

*proof.* In our key issuing model, a user and the KGC may collude to compute any private key of another user. In this case, since the KGC can compute any STK whichever it wants, forging the LTK is sufficient to compute a private key of a user. However, by the Assumption 5.1  $\sim$  5.3, it is computationally infeasible for a user to compute any LTK or STK corresponding to  $\langle \text{ID}, T \rangle$  pair. More specifically, to obtain a LTK, the KGC and a user can try to do the following using the publicly available information: 1) extract LTK, 2) extract the master key of KPAs, and 3) collect more than or equal to  $t$  LTK shares.

However, since all the following are infeasible by the Assumption 5.2, it is infeasible for a user or the KGC to acquire any information that results in a legal LTK.

- To extract  $x$  from  $P_{\text{KPAs}} = xP$  or any  $x_i$  from  $P_{\text{KPA}_i} = x_i P$ , even though  $P$  is a public

value.

- To extract the  $d_{\text{ID}}^{(i)}$  from  $b_i d_{\text{ID}}^{(i)} = b_i x_i Q_{\text{ID}}$ . This is because extracting  $b_i$  from  $\langle b_i P, b_i Q_{\text{ID}} \rangle$  is infeasible, even though  $P$  and  $Q_{\text{ID}}$  are public.
- To extract  $d_{\text{ID}}$  from  $b d_{\text{ID}}$ . This is because to compute  $b$  from  $\langle b P_{\text{KPAs}}, b Q_{\langle \text{ID}, T \rangle} \rangle$  or  $sZ = b d_{\langle \text{ID}, T \rangle}$  is infeasible, even though  $Q_{\langle \text{ID}, T \rangle}$ ,  $P_{\text{KPAs}}$ , and  $d_{\langle \text{ID}, T \rangle}$  are already known to the adversaries, a user and the KGC.
- To extract  $x$  for a user, who is an adversary trying to forge a LTK corresponding to  $\text{ID}'$ , from his/her own LTK  $d_{\text{ID}} = x Q_{\text{ID}}$ , although  $Q_{\text{ID}}$  is already known.

The KGC may try to forge a LTK corresponding to an ID or a private key of a user corresponding to  $\langle \text{ID}, T \rangle$  pair solely. However this is infeasible because it is infeasible when a user and the KGC collude. A user may try to forge any LTK, STK, or a private key corresponding to  $\langle \text{ID}, T \rangle$  on his/her own. However, a user cannot compute any LTK or private keys of other users, because it is also infeasible when a user and the KGC collude. Finally, a user may try to compute the STK from  $\langle \text{ID}, T \rangle$  using public information. However, this is impossible because followings are infeasible by Assumption 5.2.

- To extract  $s$  from  $P_{\text{KGC}} = sP$ , even though  $P$  is public.
- To compute  $d_{\langle \text{ID}, T \rangle}$  from  $sZ = b d_{\langle \text{ID}, T \rangle}$ . Because it is infeasible to extract  $b$  from  $\langle b P_{\text{KPAs}}, b Q_{\langle \text{ID}, T \rangle} \rangle$ , although a user already knows  $P_{\text{KPAs}}$  and  $Q_{\langle \text{ID}, T \rangle}$ .
- To extract  $s$  for a user, who is an adversary trying to forge a STK corresponding to  $\langle \text{ID}', T \rangle$ , from his/her own STK  $d_{\langle \text{ID}, T \rangle} = s Q_{\langle \text{ID}, T \rangle}$ , even though  $Q_{\langle \text{ID}, T \rangle}$  is already known.

Additionally, the adversary, malicious user, may try to compute  $x\bar{V}$  using  $\bar{V}, P_{\text{KPA}_i} = x_i P$  and  $P_{\text{KPAs}} = xP$  in order to escrow STK from verifiably encrypted STK. However,

this is also infeasible by Assumption 5.2.

Therefore, LTK and STK cannot be forged by any other entities except more than or equal to  $t$  KPAs and KGC respectively, and unforgeability property holds. Furthermore, a private key of a user is only known to its legitimate owner assuming that more than or equal to  $t$  KPAs do not collude, and privacy property holds. As a result, the lemma holds.  $\square$

**Theorem 5.1.** The advanced key issuing model satisfies all of our security requirements against every possible adversary models.

*proof.* By Lemma 5.1 and 5.2, the key issuing model satisfies every security requirements. Therefore, the theorem holds.  $\square$

### 5.3 Security against Private Key Leakage

In our scheme, a private key of a user is a bit concatenation of LTK and STK. As our scheme exploits simple blinding technique in order to provide secure channel between users and authorities, each secret value is only known to each responsible authority and appropriate user. In this reason, even though one of private keys of a user is revealed, the other keys of the user will be safe. Therefore an adversary, who knows a private key of a user  $D_{\langle \text{ID}, T \rangle} = d_{\text{ID}} + d_{\langle \text{ID}, T \rangle}$ , knows neither LTK  $d_{\text{ID}}$  nor STK  $d_{\langle \text{ID}, T \rangle}$  and cannot compute other private keys of the user,  $D_{\langle \text{ID}, T' \rangle} = d_{\text{ID}} + d_{\langle \text{ID}, T' \rangle}$ . However, if the adversary collude with the KGC, they can compute other private keys of the user. This is because the KGC can compute any STK  $d_{\langle \text{ID}, T \rangle}$  it wants. However, as the KGC is trusted authority, the probability that this kind of attack happens is negligible.

## Chapter 6

# Conclusion

In this paper, we have proposed a new efficient model for issuing private keys in IBE scheme based on the Weil pairing. In our scheme, a private key is constructed using two values that are issued by two different parties, namely KPAs and the KGC. The KPAs issue LTK using the identity string of the user in a threshold manner, and the KGC issues STK using the same information used to construct the LTK plus some dynamic information.

When issuing the initial private key, our scheme provides similar level of efficiency to other schemes. However, reissuing the private key by changing the dynamic information is much more efficient than others. Therefore, our scheme is especially better than other schemes when the user requires multiple private key. In addition to this, we have eliminated the secure channel assumption by using a simple blinding technique. We can also adapt Gentry's time-slot based private key revocation approach more efficiently than others. We believe that this would be a proper solution to solve the private key revocation problem in IBCs. This is due to the fact that third-party queries used in traditional PKI would offset the advantage of IBCs.

Finally, we have presented new encryption and signature schemes based on our key

issuing model. The key issuing model also can be applied to other IBCs such as hierarchical encryption, and so on. Therefore our scheme is very practical. However, the proposed key issuing protocol generates a key which has a different form from Boneh and Franklin's [2]. Even though, we believe that many primitives intended to be operated in their setting are easily converted to ours, it would be more desirable if the new secure key issuing protocol generates keys that are identical to Boneh and Franklin's.



# Bibliography

- [1] Shamir, A.: Identity-based Cryptosystems and Signature Schemes. In: Blakley, G.R., Chaum, D. (eds.): *Advances in Cryptology, Crypto 1984*. Lecture Notes in Computer Science, Vol. 196. Springer-Verlag (1985) 47–53
- [2] Boneh, D., Franklin, M.: Identity-based Encryption from Weil pairing. In: Kilian, J. (ed.): *Advances in Cryptology, Crypto 2001*. Lecture Notes in Computer Science, Vol. 2139. Springer-Verlag (2001) 213–229
- [3] Desmedt, Y., Quisquater, J.: Public-key Systems based on the Difficulty of Tampering. In: Odlyzko, A.M. (ed.): *Advances in Cryptology, Crypto 1986*. Lecture Notes in Computer Science, Vol. 263. Springer-Verlag (1987) 111–117
- [4] Tanaka, H.: A Realization Scheme for the Identity-based Cryptosystem. In: Pomerance, C. (ed.): *Advances in Cryptology, Crypto 1987*. Lecture Notes in Computer Science, Vol. 293. Springer-Verlag (1988) 341–349
- [5] Maurer, U., Yacobi, Y.: Non-interactive Public-key Cryptography. In: Davies, D.W. (ed.): *Advances in Cryptology, Crypto 1991*. Lecture Notes in Computer Science, Vol. 547. Springer-Verlag (1991) 498–507
- [6] Cocks, C.: Identity-based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.): *Proc. of the 8th IMA Conf. on Cryptography and Coding*. Lecture Notes in Computer Science, Vol. 2260. Springer-Verlag (2001) 360–363
- [7] Gentry, C.: Certificate-based Encryption and the Certificate Revocation Problem. In: Biham, E. (ed.): *Advances in Cryptology, Eurocrypt 2003*. Lecture Notes in Computer Science, Vol. 2656. Springer-Verlag (2003) 490–497

- [8] Hess, F.: Efficient Identity based Signature Schemes based on Pairings. In: Nyberg, K., Heys, H. (ed.): *Selected Areas in Cryptography, SAC 2002*, Lecture Notes in Computer Science, Vol. 2595. Springer-Verlag (2003) 310–324
- [9] Al-Riyami, S., Paterson, K.: Certificateless Public Key Cryptography. In: Lai, C. (ed.): *Advances in Cryptology, Asiacrypt 2003*. Lecture Notes in Computer Science, Vol. 2894. Springer-Verlag (2003) 452–473
- [10] Bellare, M., Holdwasser, S.: Verifiable Partial Key Escrow. *Conference on Computer and Communications Security. Proceedings of the 4th ACM Conference on Computer and Communications Security (1997)* 78–91
- [11] Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Secure Key Issuing in ID-based Cryptography. In: Montague, P., Stekette, C. (eds.): *Proc. of the 2nd Australasian Information Security Workshop, AISW 2004*. CRPIT, Vol. 32. Australian Computer Society (2004) 69–74
- [12] Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. In: Stern, J. (ed.): *Advances in Cryptology, Eurocrypt 1999*. Lecture Notes in Computer Science, Vol. 1592. Springer-Verlag (1999) 295–310



# **A New Efficient Private Key Reissuing Model for Identity-based Cryptosystems**

Donghyun Kim

Under the Supervision of Heekuck Oh  
From the Department of Computer Science and Engineering  
Graduate School of Hanyang University

## **Abstract**

The main obstacle hindering the wide deployment of IBC (Identity-Based Cryptosystem) is that the entity responsible for creating the private key can inherently decrypt any ciphertexts and forge any signatures. One obvious solution to this problem is to apply the threshold technique. However, this increases the authentication, pairing computation, and communication cost during the key issuing phase. In this paper, we propose a new efficient model for issuing multiple private keys in IBC that also alleviates the key escrow problem. In our model, the private key of a user is divided into two components, LTK (Long-Term Key) and STK (Short-Term Key), which are issued separately by different parties. The LTK is issued in a threshold manner by KPAs (Key Privacy Agency), whereas the STK is issued by a single authority called KGC (Key Generation Center). A user can efficiently obtain a new private key containing the same identity information by changing only the STK. We also give security proof of the key issuing model, and present new IBE (Identity-Based Encryption) and IBS (Identity-Based Signature) schemes based on our key issuing model.

## 감사의 글

특별한 목적 없이 단지 공부가 좋아서 대학원에 들어 온지도 어느덧 2년이 지나갔습니다. 돌이켜 보면 저의 석사 과정은 그리 순탄치 만은 않았습니다. 몸이 아플 때나 사람 관계에 힘들어질 때 또는 하고 있던 연구가 잘 진행되지 않을 때, 저는 스스로에 대하여 의심을 가지게 되었고, 올바른 길을 가고 있는가에 대하여 자문하였습니다. 그럴 때마다 저에게 힘이 되어준 모든 분들께 감사를 드립니다.

먼저 대학 4년 및 대학원 2년 과정 동안 저에게 학문적인 가르침을 주셨던 박성한 교수님, 전창호 교수님, 김한우 교수님, 허신 교수님, 이정규 교수님, 문영식 교수님, 마상백 교수님, 박성주 교수님, 도경구 교수님, 최종민 교수님, 김정선 교수님, 손진현 교수님께 감사드립니다.

또한, 부족한 시간을 쪼개서 저의 연구를 도와주시고, 날카로운 지적으로 저의 부족한 점을 일깨워 주신 김상진 교수님께 감사드립니다. 힘들었던 시절 많은 대화를 나누며 저를 위로해 주시고 도와주신 종영이 형께 감사드립니다. 스스로 작아질 때마다 항상 위로의 말을 해 주었던 지현이 형과 혜영이에게 고마운 마음을 전하고 싶습니다. 힘든 2년간 항상 의논을 하며 서로 의지가 되었던 태균이 형과 현범이에게도 감사드립니다. 힘든 연구실 생활을 잘 견디며 함께 즐거운 추억을 만들었던 태욱, 훈정 등의 후배님들에게도 감사드립니다. 마지막으로 즐거움과 어려움을 함께한 다른 모든 대학원 선, 후배님들께 감사의 말을 전하고 싶습니다.

사회에서 저마다 바쁘게 생활하는 친한 친구인 영수, 준호, 관성 등의 친한 고향 친구들에게 시간을 내서 자주만나지 못한 것에 대하여 미안하다고 전하고 싶

습니다. 또한, 현철, 규봉, 세진, 정섭, 성진, 원서, 인원, 은식, 윤식 등의 친한 동아리 동기들에게도 자주 만나지 못해서 미안하다고 전하고 싶습니다. 그리고 그 외의 저를 응원해 주셨던 모든 분들께 감사드립니다.

스스로 좌절하고 포기했던 수많은 순간 동안 단 한번도 저에게 실망하지 않으시고 위로해주시고 격려해 주셨던 지도 교수님인 오희국 교수님께 고개 숙여 감사드립니다. 앞으로도 끊임없이 정진하여 교수님의 기대에 어긋나지 않는 훌륭한 학자가 되도록 노력 하겠습니다.

내 평생에 걸쳐 가장 근면하시고 성실하신 분들이시며, 그 누구보다도 저를 아껴주신 아버님과 어머님께 감사드립니다. 제가 힘들었던 시절 그 누구보다도 저를 위해 걱정해 주셨던 부모님들께서 안 계셨다면 지금의 저는 없었을 것입니다. 또한 저의 가능성을 누구보다도 인정해 주며 격려해 주는 누나와 제 대신 집안 잡일을 잘 해주는 말 잘 듣는 동생 그리고 제가 사랑하는 저의 친 할머니께도 감사의 마음을 전하고 싶습니다.

마지막으로 항상 혼자만을 알던 저에게 함께 사는 세상과 사랑하는 법을 가르쳐 주시고 이 작은 결실이 이루어질 수 있도록 도와 주셨던 모든 분들께 감사드리며, 이 논문을 그 분들에게 바칩니다.